

WHAT IS CLAIMED IS:

1. A computer-readable memory medium, comprising:

a first software component adapted to create a graphical representation of an object embodied as code within the software component, wherein the code, when rendered upon a display screen as said graphical representation, comprises text and other displayable content;

an application program adapted to create the graphical representation by running the application program under an operating system; and

a second software component is adapted for drawing text, and wherein the first software component is invoked during runtime by the application program to define the visual attributes of the text, but not to draw the text, and wherein the second software component is invoked to draw the text, using only said visual attributes.
2. The memory as recited in claim 1, wherein the operating system assigns text drawing placed in a buffer and the buffered text edited prior to being drawn.
3. The memory as recited in claim 1, wherein the first software component may be used to the graphical representation of the object.
4. The memory as recited in claim 1, wherein the first software component is adapted to support undo and redo editing of the text content in the graphical representation of the object.
5. The memory as recited in claim 1, wherein the object is part of a graphical user interface associated with the application program.

T01E50" 0290/960

6. The memory as recited in claim 1, wherein the application program is written in Java programming language.

7. The memory as recited in claim 1, wherein the first and second components comprises a Java virtual machine of a Swing application program interface.

8. A method for drawing an object embedded within software code, said object comprises text and other displayable content created by an application program running under an operating system, the method comprising:

executing a first software component to create a graphical representation of the object to define the visual attributes of the object absent creation of any text attributable to the object; and

executing a second software component to draw the text attributable to the object by using the visual attributes to fetch code that is independent of the operating system.

9. The method as recited in claim 8, wherein said executing comprises creating a label upon the display absent any text.

10. The method as recited in claim 8, wherein said executing comprises creating a border upon the display absent any text within the border.

11. The method as recited in claim 8, further comprising the operating system assigning text drawing placed in a buffer and the buffered text edited prior to being drawn.

12. The method as recited in claim 8, further comprising using the first software component to draw the graphical representation of the object.

09370620.053101
TOT.E50.02907860

13. The method as recited in claim 8, further comprising the first software component supporting undo and redo editing of the text content in the graphical representation of the object.

14. The method as recited in claim 8, wherein the object is part of a graphical user interface associated with the application program.

15. The method as recited in claim 8, wherein the appearance and behavior of the object is independent of the operating system.

16. A computer-readable storage device, comprising:

a windows-based operating system;

an application program running under the operating system;

an object and text associated with the object, both created at runtime by the application program, wherein the application program is adapted for

invoking a first software component adapted to create a graphical representation of the object to define the visual attributes of the object, but not to draw the text; and

invoking a second software component adapted for fast text drawing to draw the text, using said visual attributes.

09870620.053104
FOFES0.02902860